Spring 2019

Using machine learning to predict prescription opioid misuse in patients

Jacob Huinker
huinker@iastate.edu

Using Machine Learning to Predict Prescription Opioid Misuse in Patients

By

Jacob Huinker

A Creative Component submitted to the graduate faculty in partial fulfillment of the requirements of for the degree of

MASTER OF SCIENCE

Major: Information Systems

Program of Study Committee:

Anthony M. Townsend, Major Professor

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this creative component. The Graduate College will ensure this creative component is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2019

TABLE OF CONTENTS

ACKNOWLEDGMENTS 2

ABSTRACT..... 3

INTRODUCTION 3

LITERATURE REVIEW FOR CONTEXT 6

RESEARCH MODEL……..... 12

RESEARCH METHODOLOGY…… 20

DISCUSSION……..... 33

CONCLUSION 36

REFERENCES 37

ACKNOWLEDGEMENTS

I would like to thank Marlene Youde and Dr. Colleen Kummet, two former acquaintances for giving me the courage and opportunity to go back to school and produce this research masterpiece.

I would also like to thank Dr. Zhu Zhang for teaching me a wide variety of data analytics and machine learning topics that helped inspire my research in this subject.

Lastly, I would like to thank my family and friends who have supported me though my graduate school career and my research.

Using Machine Learning to Predict Prescription Opioid Misuse in Patients

By Jacob Huinker

Abstract

This paper explores different machine learning techniques to predict prescriptions opioid misuse in Medicare and Medicaid patients in the United States. The author demonstrates careful selection of the best perceived machine learning algorithms, how to select useful features for a model, as well as explaining data cleaning and validation procedures. This work also shares how machine learning can be applied in practice, helping those affected by the prescription opioid crisis.

Introduction

In recent years, the United States has been facing a crisis regarding the misuse of prescription opioids. As a result, the epidemic has attracted many media outlets to report this ongoing issue. Prescription opioids are widely initiated by the American people, being the second highest gateway drug, behind only marijuana (Brady et al. 2016). There are roughly 19 million citizens that are introduced to the drug each year (Brady et al. 2016). Overdoses of prescription opioids have risen rapidly since 2000, as fatalities from associated with overdoses are now at 44 per day (Brady et al. 2016). Most of the deaths by overdose had to do with people taking prescription opioids and illegal substances concurrently (Brady et al. 2016). Many patients who take prescription opioids usually gain a dependence on the drug, for which they show signs for craving more of a given opioid (Brady et al. 2016). Other patients have developed a tolerance of prescription opioids, which means they need to take more of the drug in order to feel the effects of the initial dosage (Brady et al. 2016).

Even though prescription opioid abuse, misuse, and addiction are very similar to each other, there are differences between the three terms. **Misuse** refers to the use of prescription opioids outside of the directions given when prescribed to the patient (Brady et al. 2016, Vowles et al. 2015). **Abuse** refers to the use of prescription opioids for a reason that is not medical related, such as gaining a recreational high or getting a sense of euphoria (Vowles et al. 2015). **Addiction** refers to a pattern of continued use or a dependence of a prescription opioid, such as craving the drug or taking it compulsively (Vowles et al. 2015).

The reason why I chose this project is to be able to apply an IT solution to help solve a problem in the healthcare industry that would benefit a customer that I am working with, the Centers for Medicare and Medicaid Services (CMS) as well as the company that I am working for, General Dynamics Information Technology (GDIT). I have interviewed a manager for a data mining team, Colleen Kummet (2018), who currently oversees efforts being done for CMS's Chronic Conditions Warehouse (CCW), and she mentioned that the prescription opioid crisis was under CMS's radar as a problem they would like to address. Therefore, I saw this as an opportunity to add a piece to the puzzle in order to help combat the prescription opioid crisis.

Aside from knowing about CMS's wishes, previous literature has indicated a dire need of being able to detect potential signs of prescription opioid misuse in order to reallocate resources to fight the epidemic. Brady et al. (2016) has indicated that healthcare workers, such as doctors and pharmacists need to take great care in monitoring certain behaviors that are related to prescription opioid use disorders and misuse, with Cochran et al. (2017) adding that monitoring such behavior can add value to a health system. Pain management has been in the limelight in recent times and healthcare providers need to decrease the negative effects associated to the increased access to opioids, while pain is still treated in a reasonable manner (Brady et al. 2016).

Many health information systems possess a vast amount of data that has the potential to be used for increasing national efforts to fight opioid misuse and overdose (Cochran et al. 2017). Therefore, *the purpose of this study is to develop the best machine learning model in order to predict a prescription opioid overdose in Medicare patients across the United States*.

Machine learning and data mining go very closely hand in hand. In fact, Koh and Tan (2005) made a connection between machine learning and data mining, stating that data mining is an offspring of statistics, database management, and machine learning in computer science. **Data mining** refers to the practice of finding hidden or unknown patterns or trends in data (Kaur and Wasan 2006, Koh and Tan 2005). **Machine learning** is similarly referred to as finding useful patterns in data in order to answer questions of interest (Wu et al. 2010). Data mining and machine learning can also be defined as the procedure of selecting data and building models to discover patterns that weren't known previously (Koh and Tan 2005).

In light of healthcare data being valuable for combating the opioid crisis, using machine learning to predict signs of prescription opioid misuse can potentially increase value to healthcare data. Rose (2018) suggests that prediction algorithms that apply medical knowledge with machine learning tools could have a promising outcome. It has been suggested that CMS would like to use the machine learning model in order to watch over certain geographical locations (Kummet, 2018). If it's automated, it could have many potential benefits to CMS's data analytics team. Some of the advantages such an automated system could include less time and effort to the team and a reduced chance for error as opposed to conducting manual predictive modeling (Obenshain 2004). Other benefits could include correctly formatted and presentable data as well as the ability to use the results from the machine learning model in multiple areas at the same time (Obenshain 2004).

The following research questions are examined in this study:

1) *What is the best algorithm to use in order to create a model that will predict a prescription opioid overdose in patients with the highest accuracy?*
2) *What are the most meaningful features that will help the machine learning model achieve the highest accuracy?*

In this study, I will start by conducting a review of previous literature that has studied various machine learning algorithms, focusing on classification algorithms as well as providing a background on Regression, Clustering using k-Means and Association using the Apriori algorithm. Once the literature review has been conducted, I will proceed to describe my research model, which includes the choice of the top four algorithms I will use in the empirical study of choosing the best model as well as explaining which features could convey the highest meaning. After that, I will proceed to explain the methodology of my research as well as showcase the results. I will finish the study by sharing the implications and concluding the paper.

Literature Review for Context

I have been blessed to find several pieces of literature that covers many of the machine learning algorithms in considerable detail. I have also found a few instances that included several suggestions of which features to use based on previous studies on classifying prescription opioid misuse as well as literature that gives recommendations on how to select meaningful features. The literature map for this review can be found on Table 1. For the classifier algorithms, I will cover information of Decision Trees, Artificial Neural Networks (ANN), Support Vector Machines (SVM), and k-Nearest Neighbor (kNN).

Algorithm Studied	Tomar & Agarwal (2013)	Kaur & Wasan (2006)	Wu et al. (2010)	Koh & Tan (2005)	Tzeng et al. (2004)
Decision Tree	X	X		X	
Artificial Neural Networks (ANN)	X	X			
k-Nearest Neighbor (kNN)	X				
Naïve Bayes	X				
Support Vector Machine (SVM)	X		X		X
Linear Regression	X				
Logistic Regression	X				
Clustering (k-Means)	X				
Association (Apriori)	X			X	

Table 1: Literature Review Map

Decision Tree

The Decision Tree is a classification algorithm that sorts variables or features like a tree-shaped graph (Tomar and Agarwal 2013). It represents knowledge in the form of nodes and branches, giving it an appearance of a tree (Kaur and Wasan 2006). Decision Trees are similar to a flowchart in that every branch node of the tree conducts a test on each feature (Tomar and Agarwal 2013). Each end node, called a leaf node contains the class label based on the test done in the branches (Tomar and Agarwal 2013). The nodes at the top of the tree are called root nodes (Tomar and Agarwal 2013). Decision Trees work by pushing instances down the tree, where variable values match each other. (Kaur and Wasan 2006). This happens until the leaf node is reached and the class label is given (Kaur and Wasan 2006). There are several Decision Tree algorithms available with slight variations to each one of them. The algorithms include: HUNTS algorithm (original), CART, ID3, C4.5, SLIQ, and SPRINT (Kaur and Wasan 2006). Decision Trees are commonly used in operations research analysis where they are known for calculating conditional probabilities (Tomar and Agarwal 2013). Decision makers can choose the best

solution and the path from the root to the leaf means that there's a well-separated class value using the highest information gain (Tomar and Agarwal 2013).

Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) works in a way that's similar to an animal's nervous system, that uses a multitude of processing element's, more commonly known as neurons to conduct problem solving (Tomar and Agarwal 2013, Kaur and Wasan 2006). The analytical techniques in an ANN work in the same way as animals learn using cognition from neurological functions in the brain (Kaur and Wasan 2006). As a result, the algorithm is able to predict fresh observations based on previous encounters (Kaur and Wasan 2006). During the learning process, the initial rules are extracted from the previously learned network in order to improve the interoperability (Tomar and Agarwal 2013). ANNs are used for classification purposes because it is able to recognize certain patterns in data (Tomar and Agarwal 2013). Since the ANN is capable of constantly improving itself as it goes through more data, it can easily adapt to new changes by adjusting its weight to reduce error (Tomar and Agarwal 2013). The adaptive nature of ANNs make it an exceptional machine learning algorithm to work with.

Support Vector Machine (SVM)

Support Vector Machines (SVM) are unique in that it takes all points on a lower dimensional plane and moves them to a higher dimensional space, called a hyperplane (Tomar and Agarwal 2013). Alternatively, the hyperplane can also be called a "feature space" (Wu et al. 2010). Now and then, it can be tough to separate each data point in an original finite input space, so in turn, the data points are mapped to the hyperplane where each data point is separated further in the higher dimensional space (Tomar and Agarwal 2013). Since the separation of each

data point is maximized by constructing a hyperplane, it can serve as an advantage by being able to classify each data point more easily (Tomar and Agarwal 2013, Wu et al. 2010). Once the SVM is finished, the decision bounder will be non-linear when it's placed back into its original input space (Wu et al. 2010). SVMs were originally used for binary classification, but it is now also used for multiclass problems (Tomar and Agarwal 2013).

k-Nearest Neighbor (kNN)

The k-Nearest Neighbor (kNN) algorithm is a very simple classifier and it may be one of the simplest (Tomar and Agarwal 2013). kNN looks for non-identified data points by looking to its neighbors or data points it already knows for information on where to classify the new point (Tomar and Agarwal 2013). Once it gets enough information from said neighbors, it can then successfully classify the new point (Tomar and Agarwal 2013). The nice part about kNN is that is can classify each data point using multiple neighbors (Tomar and Agarwal 2013).

Naïve Bayes

Naïve Bayes is a relatively simple classifier that uses Bayes Theorem (Tomar and Agarwal 2013). Bayes Theorem is a classification algorithm that focuses on prior items to determine the probability of a new items that come in (Tomar and Agarwal 2013). In other words, Naïve Bayes looks at the data point and tries to learn about that point based on previous data points that it classified.

Aside from sharing just classification algorithms, I thought I would also share some information on a couple of regression algorithms. The two algorithms are linear regression and logistic regression.

Linear Regression

Linear Regression is relatively simple, as it seeks a relationship between and independent variable and a dependent variable (Tomar and Agarwal 2013). The algorithm is based on the linear function in mathematics where it tries to find a line (Tomar and Agarwal 2013). The algorithm calculates the vertical distances and finds the sum of least squares (Tomar and Agarwal 2013). With Linear Regression, the variables are already known and it basically tries to find a correlation between the two variables by finding a line (Tomar and Agarwal 2013). As a result, one major downfall of Linear Regression is that it can only work with numerical data.

Logistic Regression

Logistic Regression is a form of non-linear regression that uses the logit function (Tomar and Agarwal 2013). Unlike Linear Regression, Logistic Regression can predict the probability of an occurrence in categorical variables (Tomar and Agarwal 2013). There are two types of logistic regression: binomial and multinomial. Binomial Regression predicts the class based on only two possible values, such as a 0 or a 1 (Tomar and Agarwal 2013). On the other hand, Multinomial Regression can predict the class based on more than two values such as 1, 2, 3, 4, or 5 (Tomar and Agarwal 2013).

Clustering (k-Means and k-Medioids)

Clustering algorithms are quite different from classification and regression algorithms. Where classification and regression algorithms are under the category of supervised learning, clustering algorithms are under the category of unsupervised learning.

The goal of Clustering is to group different objects by similarity, creating clusters (Koh and Tan 2005). Each cluster or group, “k”, can have multiple data points “n” separated into them (Tomar and Agarwal 2013). Although one cluster can have multiple data points, each data points can only be part of one cluster (Tomar and Agarwal 2013). The two most common clustering algorithms are k-Means, where the centroid of each cluster is based on the average of values of each point, and k-Medioids, where the centroid of each cluster is based on the median of values of each point (Tomar and Agarwal 2013). With k-Means and k-Medioids, the analyst will have to specify the number of clusters before proceeding to partition the dataset into different groups (Tomar and Agarwal 2013).

Association (Apriori Algorithm)

Association is based on finding out which variables belong together (Koh and Tan 2005). The inputs used to find which variables belong together are support and confidence, which helps separate out the frequent variables from the infrequent variables (Tomar and Agarwal 2013). In particular, the Apriori algorithm, which is a popular association algorithm checks for variables that are used frequently versus variables that are used infrequently (Tomar and Agarwal 2013). If the variable doesn’t meet a certain threshold of frequency, the algorithm proceeds to cut the variable out as it doesn’t see the variable as making a contribution to the association rules (Tomar and Agarwal 2013).

The literature review process has helped deliver the context of several machine learning algorithms. The studies of previous literature has helped me shape my decision of choosing the top four algorithms based on the feedback that was given. I will share these four algorithms in

the next section, which covers my research model. I will also share the suggested variables based on previous literature as well.

Research Model

After reviewing the advantages and disadvantages of each machine learning algorithm, I have chosen the top five algorithms to further my empirical study on. The algorithms are Decision Tree, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Logistic Regression, and Association using the Apriori algorithm. I will explain my rationale for choosing the three algorithms using support from the literature that was reviewed. After selecting each algorithm, I will proceed to explain the potential features I will use, given information from previous literature. I will end the research model section with a note on what I found to be a potentially good approach to validating each model.

Decision Tree

The reason why the Decision Tree was chosen first was because of its many advantages. Since the Decision Tree doesn't require a lot of computational expense to construct, it is easy for analysts to understand, and it can easily be integrated with a database if an institution wishes to do so (Kaur and Wasan 2006). Due to rule induction, the Decision Tree can be used to easily classify new class cases (Kaur and Wasan 2006). Aside from easy understanding, the Decision Tree is also visually appealing to many analysts (Koh and Tan 2005). Aside from SVMs, the Decision Tree can also process higher dimensional data as well, since it is capable of assigning exact class values no matter the complexity of the data (Tomar & Agarwal 2013). The Decision Tree can also handle data that possesses both numerical and categorical properties, which is good for large healthcare data sets (Tomar and Agarwal 2013). Because of these benefits, Decision

Trees are used extensively in healthcare research (Tomar and Agarwal 2013). Along with wonderful advantages of using a Decision Tree, there are also disadvantages. A couple of workable disadvantages is the fact that the Decision Tree is limited to using only one class variable and that the class can only be a categorical variable (Tomar and Agarwal 2013). This is okay since this study will be trying to solve a simple classification problem of predicting whether a patient is misusing prescription opioids or not. The Decision Tree can potentially be unstable depending on the type of data set that is used (Tomar and Agarwal 2013). Even though this cannot be easily mitigated, data cleaning and preparation should help alleviate instability to a certain extent.

Artificial Neural Network (ANN)

The next algorithm of choice is the Artificial Neural Network (ANN) because of its advantageous performance in previous healthcare applications. ANNs are very flexible in terms of being capable to perform clustering and generate predictive models (Kaur and Wasan 2006). They are able to identify relationships between the independent variable and the class variable with a high accuracy performance (Tomar and Agarwal 2013, Kaur and Wasan 2006). One of the possible reasons behind this is the ANN's ability to handle noise, which is common in large datasets (Tomar and Agarwal 2013). All of these advantages make the ANN a viable decision assistant in the healthcare industry. A couple of major disadvantages is the complexity of the finished ANN, which can be difficult for analysts to understand (Tomar and Agarwal 2013). This could be due to the fact that they blindly find relationships between dependent and independent variables, making ANNs unable to explain why it made the connection (Kaur and Wasan 2006). Although ANNs are notorious for their outstanding accuracy, it also introduces the possibility that the algorithm could over-fit the data (Tomar and Agarwal 2013). The disadvantages can be

looked past since the goal of this study is to find an algorithm that will achieve the highest amount of correctly classified instances.

Support Vector Machine (SVM)

Like the Decision Tree and Artificial Neural Network (ANN), Support Vector Machines (SVM) have ideal features that makes it a powerhouse to use as a machine learning algorithm. One of the most prominent features of SVMs is the fact that the algorithm has a promising empirical performance (Tomar and Agarwal 2013). Like ANNs, SVMs produce very high accuracies, usually almost 100 percent once a model has been optimized (Tzang et al. 2004). Due to the SVM's capability to move each data point into a higher dimension, it can handle data points with higher complexities (Tomar and Agarwal 2013). Unlike ANNs, SVMs doesn't have as much of an issue of overfitting (Tomar and Agarwal 2013). All of these advantages make SVMs a popular choice as a classification algorithm among healthcare researchers (Tomar and Agarwal 2013). One major disadvantage of SVMs is the fact that they are computationally expensive and that they can take quite a bit of time to process training data than the Decision Tree and the ANN (Tomar and Agarwal 2013). SVMs were originally designed to solve binary class problems, even though it is capable of solving multiclass problems (Tomar and Agarwal 2013). It solves multi-class problems by breaking them down into pairs (Tomar and Agarwal 2013). This is fine in our instance, but if we were using SVMs to solve to determine a class based on multinomial levels of severity of prescription opioid misuse, the SVM would take quite a while due to its complex way of processing. Other disadvantages include the analyst having to choose a kernel function which could affect the performance of the SVM model (Tomar and Agarwal 2013).

Logistic Regression

Even though previous literature hasn't touched a whole lot on the advantages and disadvantages Logistic Regression, I thought the features explained in the literature review was enough to give this regression algorithm a chance. Regression models in general are known for their highly predictive properties. As stated in the literature review, Logistic Regression algorithms are capable processing categorical variables, something that is common as well as numerical variables in healthcare data (Tomar and Agarwal 2013). This serves as a major advantage in the light of our goal being to be able to correctly predict a prescription opioid overdose in patients. Logistic Regression has also been used for prediction in the healthcare field such as predicting the survivability of patients (Tomar and Agarwal 2013). A medical surveillance system, the Real-Time Outbreak and Disease Surveillance system (RODS), used a Recursive Least Squares (RLS) algorithm, one that's related to Logistic Regression, to detect disease outbreaks (Espino et al. 2004).

Others – Not Selected

This sub-section will explain both the advantages and the disadvantages of each machine learning algorithm that are although good ones to use, aren't seen in this light as producing as strong of results as the top three. The honorable mentions include k-Nearest Neighbor (kNN), Naïve Bayes, Linear Regression, and, Clustering.

Previous literature hasn't touched a whole lot on advantages and disadvantages either, but I thought that it would be good to consider an Association algorithm, more specifically, the Apriori algorithm. However, the Apriori algorithm ranks each feature instead of guessing a binary class, so it won't be able to be tested in this study. Back in the literature review, it was

stated that Association algorithms are known for discovering which variables go together (Tomar and Agarwal 2013). With this in mind, it may be beneficial to use the Apriori algorithm in order to find the relationships between different variables (Tomar and Agarwal 2013). The advantage will come from the fact that healthcare data contains many variables that are related to each other in some sense, so it gives the Apriori algorithm the potential to perform well.

k-Nearest Neighbor (kNN) has its advantages in terms of simplicity. It's very easy to implement and training a kNN model can be done very quickly at very little computational expense (Tomar and Agarwal 2013). However, kNN is sensitive to noise, which could pose a problem when using a large healthcare data set and it requires a vast amount of storage in order to store each data point (Tomar and Agarwal 2013).

Naïve Bayes has a balance of both advantages and disadvantages in most applications. However, the disadvantages could weigh in more in the instance of working with healthcare data. Naïve Bayes classifiers are notorious for high accuracies in general IT industries, as well as faster computation when during training (Tomar and Agarwal 2013). However, the major disadvantage is the fact that Naïve Bayes assumes that all variables are independent (Tomar and Agarwal 2013). This may be fine in other settings, but this is a major drawback when working with healthcare data due to all variable having high correlations with each other. In the instance of this study, it is likely that the Naïve Bayes algorithm won't perform as well as the top three selected algorithms.

Linear Regression, although is great for predictive modeling, has a major disadvantage when it comes to healthcare data. Since Linear Regression works with only numerical data and not categorical data, it will make a creation of a model using this algorithm impossible since healthcare data has both numerical and categorical values (Tomar and Agarwal 2013).

Clustering works great in many machine learning and data mining applications, but it has its drawbacks as well. Like kNN, Clustering algorithms such as k-Means are simple and efficient, and requires less computational expense to train the model (Tomar and Agarwal 2013). However, Clustering algorithms have trouble clustering data points with categorical variables, which is a major disadvantage when working with healthcare data (Tomar and Agarwal 2013). Clustering is best used when an analyst knows less about a dataset (Tomar and Agarwal 2013). Although not as severe, Clustering algorithms also require a set number of clusters (Tomar and Agarwal 2013).

Feature Selection

Aside from selecting algorithms to use in a machine learning model, it is also very important to select the appropriate features or variables that will give the most meaning to the model. Tomar and Agarwal (2013) stated that data analysts need to recognize variables that would be considered inappropriate since irrelevant variables can act as noise. This in turn, can disrupt or even slow the machine learning process (Tomar and Agarwal 2013). Originally, I had planned on using Principal Component Analysis (PCA), but I have discovered from past literature that it is recommended that it shouldn't be used in the context of a health care setting. Verma et al. (2013) explains that dimensionality reduction methods do not work so well on the interoperability of results when applied to healthcare data. Since the Principal Component Analysis falls under the category of a dimensionality reducing feature selection algorithm, it cannot be used (Verma et al. 2013). Instead, I have decided to use the Gini Index in order to select features for the Decision Tree and Artificial Neural Networks (ANN) algorithm, and use L1-norm penalized coefficients for the Support Vector Machine (SVM) algorithm. The Gini Index uses a range between 0 and 1, where 0 indicates maximum information gain for a feature

and 1 indicates no information gain for a feature (Verma et al. 2013). Therefore, features that have a Gini Index that is closer to 0 will have a higher chance of being selected as opposed to a feature that has a Gini Index that is closer to 1. Wu et al. (2010) chose Bayesian Information Criterion (BIC) for their model selection algorithm and they decided to go with L1- norm penalized variable selection, a note that will be kept in mind for the empirical study that will be performed later in this paper. For the sake of scope and project time, I will just be doing feature selection and will refrain from performing model selection tasks.

After reading literature on previous studies regarding detecting prescription opioid misuse, I have learned about a plethora of different features that could possibly be used in the study later in this paper. Demographics such as age, gender, ethnicity, income from security assistance benefits, urban or rural living location, history of sexual assault and violence, and mood and anxiety disorders all are potentially important features to be used in my study (Brady et al. 2016, Cochran et al. 2017, Vowles et al. 2015). Opioid prescription and use history, such as prescription history, the route of drug administration, any reasons for escalation will be equally as important (Brady et al. 2016, Vowles et al. 2015). Information on pain and its management will also be important, such as a patient's age at onset, the duration of said pain, pain location, and the history of treatment (Brady et al. 2016, Vowles et al. 2015). Other potentially useful features include information on the use of other opioids and substances outside of prescription opioids such as illegal substances and alcohol use are not ruled out (Brady et al. 2016, Cochran et al. 2017). The frequency of emergency department visits will also be another important feature to consider (Cochran et al. 2017).

Model Validation

Even though selecting the correct algorithm and features are very important, it is also important to come up with an approach to validating the completed model. Most analytics practitioners use a classic training data set and a test data set. The **training data** is a larger set of data used by classification algorithms to analyze and learn the various properties of the data in order to create a working model (Kaur and Wasan 2006). The **test data** is a smaller set of data used to assess the model and to estimate the accuracy (Kaur and Wasan 2006, Obenshain 2004). Tzeng et al. (2004) used this classic approach to train and validate their SVM model.

Aside from the classic training and test data approach, I thought that I could try using k-Fold Cross Validation. Rose (2018) suggested that k-Fold Cross Validation is an accepted standard that should be adopted in healthcare machine learning applications, aside from the classic training and test data samples. K-Fold cross validation is when the data being analyzed is split into k mutually exclusive data sets (Rose, 2018). The chosen k data set will be withheld as a validation set and the other non-chosen data sets will be used to train the model. One perk of using k-Fold Cross Validation is that the predicted values could assess overfitting with more effectiveness as well as have less variance (Rose 2018). Tomar and Agarwal (2013) also mentioned that using k-Fold Cross Validation can help improve the success of each model by using every instance of data for both training and testing.

The review and analysis of literature has helped pave the way for shaping what has been determined to be most appropriate algorithms and features to use in order to produce the best model for predicting prescription opioid misuse. In order to see which proposed model performs the best, I will now explain my research methodology, which includes the background of a secondary data set that I used, the procedure of selecting the best features, and most importantly, the procedure of training each algorithm and validation.

Research Methodology

Data

The data I am working with is a secondary data set that is publicly available from the Centers for Medicare and Medicare Services (2013). The dataset is split into several large Excel files that can be easily joined together. The files contain CMS beneficiary claims data involving inpatient visits and outpatient visits spanning from 2008 to 2010. There are 66,000 instances of inpatient claims, 790,000 instances outpatient claims, and over 1 million instances of prescription drug event data. Inpatient claims refer to longer term visits such as hospital stays. Outpatient claims refer to shorter term visits such as an emergency room visit or a same day surgery. In both the inpatient and outpatient claims data, there is an extensive set of International Classification of Disease, 9th Edition (ICD-9) code columns that will be used to create the class variable column. The class variable will be a 1 if any one of the ICD-9 codes related to prescription opioid overdose or abuse. A value of 0 will be given to the class variable if the ICD-9 code doesn't have any codes related to prescription opioid overdose or abuse. Each data file is a comma separated value (.csv) file and was able to be edited with Microsoft Excel 2013. The list of variables along with their meanings can be found on Table 2 below.

Variable (Attribute) Name	Meaning
DESYNPUF_ID	Beneficiary Code
BENE_BIRTH_DT	Date of Birth
BENE_DEATH_DT	Date of Death
BENE_SEX_IDENT_CD	Sex
BENE_RACE_CD	Beneficiary Race Code
BENE_ESRD_IND	End Stage Renal Disease Indicator
SP_STATE_CD	State Code
BENE_COUNTY_CD	County Code

BENE_HI_CVRAGE_TOT_MONS	Total Number of Months of Part A Coverage for the Beneficiary
BENE_SMI_CVRAGE_TOT_MONS	Total Number of Months of Part B Coverage for the Beneficiary
BENE_HMO_CVRAGE_TOT_MONS	Total Number of Months of HMO Coverage for the Beneficiary
PLAN_CVRG_MOS_NUM	Total Number of Months of Part D Plan Coverage for the Beneficiary
SP_ALZHDMTA	Chronic Condition: Alzheimer or Related Disorders or Senile
SP_CHF	Chronic Condition: Heart Failure
SP_CHRNKIDN	Chronic Condition: Chronic Kidney Disease
SP_CNCR	Chronic Condition: Cancer
SP_COPD	Chronic Condition: Chronic Obstructive Pulmonary Disease
SP_DPRESSN	Chronic Condition: Depression
SP_DIABETES	Chronic Condition: Diabetes
SP_ISCHMCHT	Chronic Condition: Ischemic Heart Disease
SP_OSTEOPRS	Chronic Condition: Osteoporosis
SP_RA_OA	Chronic Condition: Rheumatoid Arthritis and Osteoarthritis (RA/OA)
SP_STRKETIA	Chronic Condition: Stroke/Transient Ischemic Attack
MEDREIMB_IP	Inpatient Annual Medicare Reimbursement Amount
BENRES_IP	Inpatient Annual Beneficiary Responsibility Amount
PPPYMT_IP	Inpatient Annual Primary Payer Reimbursement Amount
MEDREIMB_OP	Outpatient Institutional Annual Medicare Reimbursement Amount
BENRES_OP	Outpatient Institutional Annual Beneficiary Responsibility Amount
PPPYMT_OP	Outpatient Institutional Annual Primary Payer Reimbursement Amount
MEDREIMB_CAR	Carrier Annual Medicare Reimbursement Amount
BENRES_CAR	Carrier Annual Beneficiary Responsibility Amount
PPPYMT_CAR	Carrier Annual Primary Payer Reimbursement Amount
CLM_ID	Claim ID
SEGMENT	Claim Line Segment
CLM_FROM_DT	Claims Start Date
CLM_THRU_DT	Claims End Date
PRVDR_NUM	Provider Institution
CLM_PMT_AMT	Claim Payment Amount
NCH_PRMRY_PYR_CLM_PD_AMT	NCH Primary Payer Claim Paid Amount
AT_PHYSN_NPI	Attending Physician – National Provider Identifier Number
OP_PHYSN_NPI	Operating Physician – National Provider Identifier Number
OT_PHYSN_NPI	Other Physician – National Provider Identifier Number
CLM_ADMSN_DT	Inpatient Admission Date

ADMTNG_ICD9_DGNS_CD	Claim Admitting Diagnosis Code
CLM_PASS_THRU_PER_DIEM_AMT	Claim Pass Thru Per Diem Amount
NCH_BENE_IP_DDCTBL_AMT	NCH Beneficiary Inpatient Deductible Amount
NCH_BENE_PTA_COINSRNC_LBLTY_AM	NCH Beneficiary Part A Coinsurance Liability Amount
NCH_BENE_BLOOD_DDCTBL_LBLTY_AM	NCH Beneficiary Blood Deductible Liability Amount
CLM_UTLZTN_DAY_CNT	Claim Utilization Day Count
NCH_BENE_DSCHRG_DT	Inpatient Discharged Date
CLM_DRG_CD	Claim Diagnosis Related Group Code
ICD9_DGNS_CD_1 – ICD9_DGNS_CD_10	Claim Diagnosis Code 1 – Claim Diagnosis Code 10
ICD9_PRCDR_CD_1 – ICD9_PRCDR_CD_6	Claim Procedure Code 1 – Claim Procedure Code 6
HCPCS_CD_1 – HCPCS_CD_45	Revenue Center HCFA Common Procedure Coding System 1 – Revenue Center HCFA Common Procedure Coding System 45
NCH_BENE_PTB_DDCTBL_AMT	NCH Beneficiary Part B Deductible Amount
NCH_BENE_PTB_COINSRNC_AMT	NCH Beneficiary Part B Coinsurance Amount
PDE_ID	CCW Part D Event Number
SRVC_DT	RX Service Date
PROD_SRVC_ID	Product Service ID
QTY_DSPNSD_NUM	Quantity Dispensed
DAYS_SUPLY_NUM	Days Supply
PTNT_PAY_AMT	Patient Pay Amount
TOT_RX_CST_AMT	Gross Drug Cost

Table 2: List of Variables (Attributes) in Data Set (Centers for Medicare and Medicaid Services 2013)

Data Cleaning and Preparation

The first thing that needed to happen before machine learning takes place was the process of cleaning and preparing the data. A few pieces of previous literature has noted the importance of data cleaning and that not cleaning and preparing the data beforehand can be disruptive in the machine learning process. Tomar and Agarwal (2013) noted that having high quality data that is also relevant is one of the biggest challenges when mining healthcare data.

I started by cleaning each raw data file in Excel. In the inpatient and outpatient claims data, it has been discovered that the ICD-9 codes are not in the correct format. It turns out that none of the ICD-9 codes have a separator period after the 3rd digit (after the 1st and 4th digits for ICD-9 codes that start with an E). Since the lengths and formats of ICD-9 codes are fairly consistent otherwise, I was able to work with each ICD-9 code by simply not adding a period that would otherwise serve as a separator. Another issue I found is that there were null values, which created a barrier to uploading each data set to the database. I resolved this issue by temporarily replacing null values with zeroes. Once the data was uploaded, I put the null values back in the text-based fields and kept the zeroes for numeric-based fields. The claims beneficiary data was in three separate years when the inpatient claims and outpatient claims contained data for all three years (2008-2010). As a result, I merged the beneficiary data files for each year into one year that contained all three years of data. Since the size for each of the beneficiary files were relatively small and the columns were identical, is was fairly easy to do a simple copy and paste. Aside from the larger issues, I also cleaned up any inconsistent data values that arose.

The next preparation task was to join all of the claims data files into one large data file via a database. Due to previous knowledge on Oracle databases, I have decided to use a MySQL database, a free open source database provided by Oracle. In order to import each data file with as little risk as possible, I decided to use MySQL's built in data import wizard to import each .csv file. The wizard consisted of naming the new table, choosing the correct data types for each column, and starting the import process. As mentioned before, databases can be picky I needed to make sure all null values as well as any inconsistent data was fixed before the import started. Due to the sheer volume of each data file, it took a considerable amount of time to import them all, about 3 weeks in total.

One that was completed, I wrote a SQL query to merge all tables into one super table. I started by merging the inpatient and outpatient claims together, via a union join. That was nested inside a join with the prescription drug events and the beneficiary summary tables. A note to make is that there were duplicate rows for the beneficiary summary and claims data. This was because a beneficiary could have more than one inpatient or outpatient stay, and each stay could involve having more than one prescription drug being given to them.

In order to determine the class, I searched for a certain set of ICD-9 codes that indicated that a patient was diagnosed with prescription opioid poisoning or a use disorder. There were multiple ICD-9 columns, for diagnosis purposes and procedural purposes. I ended up using both types in order to obtain the largest amount of positive cases of misuse possible. As a part of my SQL query, I created a misuse class variable column to indicate misuse of prescription opioids. The column was populated with a 1 if there was an indication of prescription opioid misuse and a 0 if there wasn't any indication of a prescription opioid misuse. In order to prevent a bias during the machine learning process, I removed all ICD-9 columns from the final sample.

Aside from merging each table and creating the class variable, I also needed to exclude certain instances that would create a bias or a moral conflict when conducting machine learning. During my interview with Kummet (2018), I was told that all patient need to be 18 years of age or older due to issues with consent in minors. This issue was also brought up in previous literature as well (Cochran et al. 2017, Vowles et al. 2015). As a result, I have decided to exclude patients who are 17 years of age or younger from the sample. Another issue from previous literature was the fact that there are medical claims from people suffering from cancer or receiving hospice services (Cochran et al. 2017, Vowles et al. 2015). They cannot be in the sample due to the need for prescription opioids to reduce inevitable pain. Therefore, I have

decided to exclude anyone with a cancer diagnosis code from the sample as well as exclude anyone with an indication that they are in hospice. I also excluded patients who are receiving long term care for 90 or more days due to a more supervised administration of prescription opioids (Cochran et al. 2017). Another factor that I took into consideration are beneficiaries with suicide diagnoses, indicating that they may have intentionally overdosed on prescription opioids. As a result, I have also excluded them from the final sample.

One challenge I ran into was the difficulty to bring back a robust sample of data using the MySQL database. I ran into problems where it was taking the database engine several days to return queries, especially if a large number of rows were requested. This was most likely attributed to the sheer volume of each table being merged as well as the complexity of the SQL query itself. It didn't help that there were a large number of ICD-9 codes that were being used as part of the exclusion criteria and they were being stored on separate reference tables. However, I did manage to bring back a small dataset that provided sufficient information to carry on with the experiment, but it only has about 10 percent of the expected number of instances.

One issue that is pondered is balancing the dataset in order to make the class values more proportional to each other. Wu et al. (2010) suggested that machine learning algorithms are developed assuming that the data given to it is balanced. They argued that under-sampling the majority class value while keeping the minority class value the same size may benefit the machine learning process (Wu et al. 2010). I initially intended to use the full data set to simulate a real-world scenario. However, due to the difficulty in learning a vastly disproportionate spread between a positive and negative class, I have decided to balance the data to make it more proportional. The final dataset consisted of 2/3 of the rows having a negative class value and 1/3

of the rows having a positive class value. Once the dataset was ready, I saved it as a .csv file in order to import into my program.

Model Development

Although there are a few educational applications available for doing machine learning experiments, the potential size and complexity of the dataset were much too large for an educational application to handle. Therefore, I decided to hand-make a program using pre-written libraries to perform my machine learning experiments. My homemade program was written in Python 3.5 using the Sci-Kit Learn library, written by Pedregosa et al. (2011). In order to accommodate Sci-Kit Learn, I also had to implement the Numpy Library (Oliphant 2006).

I originally installed Python during the summer of 2018 and ported the language into the Eclipse Oxygen using an add-on called PyDev. This helped me code my program with some assistance that a plain code editor could not offer. I also installed the Numpy Library (Oliphant 2006) during the same timeframe. I installed Sci-Kit Learn (Pedregosa et al. 2011) in December 2018 when I was browsing and selecting most favorable library to use. Once everything was installed and set up, I started coding my program.

To start coding, I imported Numpy (Oliphant 2006) and the Sci-Kit Learn (Pedregosa et al. 2011) libraries that I needed to run each algorithm and function. I also imported Python's CSV and Excel libraries to that I could read .csv files and write out to Excel files. The first section that was coded was the process of reading in the .csv data file and storing into a Numpy array (Oliphant 2006). For my feature selection section, I needed to somehow import the names separately from the rest of the dataset. I remedied this by reading the file in two separate functions, one taking in the first row that just extracted the column names, and another function

that parsed in the body of the file. Once the data was imported, I added some code to randomize rows of the dataset since my raw file had the classes separated evenly by hand. That way, there would not be any bias from a pattern created by the user that may be picked up by the computer. One thing I learned during coding is that Sci-Kit Learn learns using data that has a certain encoding. Luckily, I was able to use a data translator under Sci-Kit Learn's preprocessing library that was able to encode the data that could be recognized by Sci-Kit Learn (Pedregosa et al. 2011). To finish pre-processing, I coded logic to split the dataset into training and testing sets respectively. In order to train and test using a reasonable amount of data instances, I decided to split the training and testing sets into 70 percent and 30 percent respectively.

Once I coded my data pre-processing tasks, I coded my feature selection algorithms. For the Decision Tree, Artificial Neural Networks (ANN) and Logistic Regression, I said I would use the Gini Index to determine the most meaningful features to use. To do this, I implemented Sci-Kit Learn's ExtraTreesClassifier() algorithm, since it contained the Gini Index to use as a solver (Pegregosa et al. 2011). I ran all data points through the classifier and printed each feature name along with their respective ranking. The next feature selection algorithm I coded was L1-norm penalized which previously mentioned, accommodated with selecting the optimal features for Support Vector Machines (SVM). To do this, I coded using the LinearSVC() algorithm, using the L1 solver. I also printed out each feature name along with their respective ranking.

After each feature selection algorithm was coded, I proceeded to code each machine learning algorithm. As a recap, the four machine learning algorithms I coded were Decision Tree, Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Logistic Regression. All four algorithms were fairly uniform in setting up, the main difference being the use of different libraries. Each algorithm took in the training set and trained the model, then took in the

testing set and tested the model. Once each model was trained and tested, I printed out each metric, including the accuracy, precision, recall, f-statistic, and the confusion matrix. Aside from implementing logic for the classic training and testing sets, I also implemented logic to perform k-Fold Cross Validation. Since k-Fold Cross Validation trains and tests on all parts of the dataset, I needed to pass the whole set to the algorithm to do a series of training and tested. To simplify the scope of testing k-Fold Cross Validation, I decided to set up the model using 10 folds. Aside from printing the results within the console, I also implemented logic to print the results to an Excel file, with each model having its own tab. The Excel file was used to store the results for analysis afterwards.

Results

Once my program was coded and tested, I started my experiment by examining the rankings of each feature. Table 3 shows the list of features in order of ranking for both the Gini Index and L1-Norm Penalized algorithms. Each ranking implies the importance of each feature as seen by each algorithm. This shows the data scientist which features are appropriate for keeping and which features can be removed. Upon analyzing the list of features, I noticed that features that indicated a chronic condition as well as the characteristics of a patient carried greater importance than administrative-related features, such as claims payments and reimbursement information.

Feature (Gini Index)	**Ranking**	**Performance and Stability Cutoff Markings**	**Feature (L1-norm Penalized)**	**Ranking**	**Performance and Stability Cutoff Markings**

SP_OSTEOPRS:	0		BENE_BIRTH_DT:	0	
SP_RA_OA:	0.000005165352405		BENE_ESRD_IND:	0	
BENE_RACE_CD:	0.00001470530655		BENE_COUNTY_CD:	0	
SP_ISCHMCHT:	0.00001665157116		PLAN_CVRG_MOS_NUM:	0	
BENE_COUNTY_CD:	0.00001782300593		SP_ALZHDMTA:	0	
SP_DIABETES:	0.00002172603042		SP_CNCR:	0	
SP_ALZHDMTA:	0.00002345895533		SP_COPD:	0	
SP_COPD:	0.00003392424922		SP_ISCHMCHT:	0	
MEDREIMB_OP:	0.00005054450393		SP_OSTEOPRS:	0	
PPPYMT_IP:	0.00005115550399		BENRES_IP:	0	
SRVC_DT:	0.0000523832851		PPPYMT_IP:	0	
BENE_ESRD_IND:	0.00006339465754		CLM_FROM_DT:	0	
PLAN_CVRG_MOS_NUM:	0.00006549752866		QTY_DSPNSD_NUM:	0	
SP_STRKETIA:	0.00006729380713		SEGMENT:	0.00000275106725	
BENE_SEX_IDENT_CD:	0.00008323862022		DAYS_SUPLY_NUM:	0.000003018894435	
SP_DEPRESSN:	0.00008953355522		SP_CHRNKIDN:	0.000003086210049	
SP_CHF:	0.00009744226544		BENE_SEX_IDENT_CD:	0.000005802839042	
PTNT_PAY_AMT:	0.000105644633		SP_STATE_CODE:	0.000006535153039	
SP_CHRNKIDN:	0.0001127551433		AT_PHYSN_NPI:	0.000009812380613	
SP_CNCR:	0.0001144879635		BENE_DEATH_DT:	0.0000138512404	
# DESYNPUF_ID:	0.0001183797258		PPPYMT_CAR:	0.00001418046041	
BENE_HI_CVRAGE_TOT_MONS:	0.0001334280037		NCH_PRMRY_PYR_CLM_PD_AMT:	0.00001897085351	
BENE_SMI_CVRAGE_TOT_MONS:	0.0001471323433		SP_STRKETIA:	0.00001981907792	
BENE_HMO_CVRAGE_TOT_MONS:	0.0001791942726		PPPYMT_OP:	0.00003259501133	
MEDREIMB_IP:	0.0002272726362		MEDREIMB_CAR:	0.00003401014647	
BENE_BIRTH_DT:	0.0002992989422	Decision Tree - Greatest Stability k-Fold Cross Validation	CLM_PMT_AMT:	0.00003764733024	
SP_STATE_CODE:	0.0003409550566	Decision Tree - Highest Performing k-Fold Cross Validation	OP_PHYSN_NPI:	0.00004357041037	

QTY_DSPNSD_NUM:	0.0003735287129		PTNT_PAY_AMT:	0.00004578776416	
TOT_RX_CST_AMT:	0.0004710611762		OT_PHYSN_NPI:	0.00005487384028	
BENRES_IP:	0.0005305604347		PROD_SRVC_ID:	0.00005909756018	
BENE_DEATH_DT:	0.0005547359396		PRVDR_NUM:	0.0001018975123	
NCH_BENE_PT_COINSRNC_AMT:	0.0007799328736		BENRES_CAR:	0.000110809435	
OT_PHYSN_NPI:	0.001949573876		NCH_BENE_BLOOD_DDCTBL_LBLTY_AM:	0.0001151395927	
CLM_PMT_AMT:	0.002090149385		BENRES_OP:	0.0001583315135	
DAYS_SUPLY_NUM:	0.002113175113		SP_RA_OA:	0.0002912753726	
SEGMENT:	0.002881414812	Decision Tree - Highest Performing Classic T&T	BENE_SMI_CVRAGE_TOT_MONS:	0.02121240778	
NCH_PRMRY_PYR_CLM_PD_AMT:	0.003071959226		MEDREIMB_OP:	0.03923005006	
CLM_THRU_DT:	0.003191038766		MEDREIMB_IP:	0.04018441718	
PPPYMT_OP:	0.003376498656		BENE_HMO_CVRAGE_TOT_MONS:	0.04661841398	
PRVDR_NUM:	0.003443443992	ANN - Greatest Stability Classic T&T	SP_DEPRESSN:	0.06872472485	
NCH_BENE_DDCTBL_AMT:	0.003701782048		BENE_RACE_CD:	0.1425966263	
CLM_FROM_DT:	0.003737440031		SP_CHF:	0.2406311033	
AT_PHYSN_NPI:	0.004812556177		# DESYNPUF_ID:	0.3394789886	
PROD_SRVC_ID:	0.004912273595		SRVC_DT:	0.4325296794	
NCH_BENE_BLOOD_DDCTBL_LBLTY_AM:	0.005910274387		BENE_HI_CVRAGE_TOT_MONS:	0.493067299	
BENRES_CAR:	0.007990426612	Decision Tree - Greatest Stability Classic T&T Logistic Regression - Highest Performing k-Fold Cross Validation Logistic Regression - Greatest Stability k-Fold Cross Validation	TOT_RX_CST_AMT:	0.5454551088	SVM - Highest Performing Classic T&T

BENRES_OP:	0.008624182761		NCH_BENE_PT_COINSRNC_AMT:	0.5932824635	
MEDREIMB_CAR:	0.009049659077	Logistic Regression-Highest Performing Classic T&T Logistic Regression-Greatest Stability Classic T&T	SP_DIABETES:	0.7751241125	SVM -Highest Performing k-Fold Cross Validation
OP_PHYSN_NPI:	0.00946265654	ANN - Highest Performing Classic T&T ANN - Greatest Stability k-Fold Cross Validation	NCH_BENE_DDCTBL_AMT:	0.9439553883	
PPPYMT_CAR:	0.01388370839	ANN - Highest Performing k-Fold Cross Validation	CLM_THRU_DT:	1.145390468	

Table 3: List of Features by Importance Ranking

After examining each list of features, I ran each algorithm to analyze each performance measure for each algorithm. Upon doing an initial run of each algorithm, I was fairly impressed by the performance of all four algorithms. However, I still needed to find the optimal performance using the optimal number of features. I started by running each algorithm using the entire set of features, then running each algorithm again by removing the least important feature, then the second least important feature, and so on and so forth. I ended up removing up to 30 of the least important features, determining that going beyond 30 features did not show any evidence of peaks in performance. I also used the same method to find the correct number of features that provided the greatest stability, meaning that each test run was consistent upon all runs for a given number of features. I made this possible by running the given set of features three times on top of removing each feature.

Table 4 shows the measures for each measure for each algorithm, displaying both the highest performing metrics as well as the metrics for the most stable runs. The table also shows a side-by-side comparison of the classic training and testing sets as well as k-Fold cross validation.

Note that many of the results were very impressive, with all four algorithms giving accuracies that were above 90 percent with the accuracy for SVM's classic training and testing sets reaching 100 percent. Another impressive observation is that k-Fold Cross Validation metrics both performed well, as delivering consistent results. This provided for greater stability without much regard for the number of features being used while maintaining modestly high performance. Many of the high performing results had higher accuracies, but were less stable, meaning that each run could produce significantly higher or lower accuracies. This was more evident with the classic training and testing runs

Algorithm	**Classic Training & Testing**		**k-Fold Cross Validation**	
	Highest Performing	**Most Stable**	**Highest Performing**	**Most Stable**
Decision Tree	Accuracy: 98.648649% Precision: 98.692241% Recall: 98.648649% F-Statistic: 98.651959% Confusion Matrix: [[43 1] [0 30]]	Accuracy: 94.594595% Precision: 94.850139% Recall: 94.594595% F-Statistic: 94.647546% Confusion Matrix: [[47 3] [1 23]]	Accuracy: 99.593496% Precision: 99.595960% Recall: 99.593496% F-Statistic: 99.592867% Confusion Matrix: [[164 0] [1 81]]	Accuracy: 99.593496% Precision: 99.595960% Recall: 99.593496% F-Statistic: 99.592867% Confusion Matrix: [[164 0] [1 81]]
Artificial Neural Networks (ANN)	Accuracy: 98.648649% Precision: 98.693694% Recall: 98.648649% F-Statistic: 98.652509% Confusion Matrix: [[44 1] [0 29]]	Accuracy: 97.297297% Precision: 97.497497% Recall: 97.297297% F-Statistic: 97.321119% Confusion Matrix: [[47 2] [0 25]]	Accuracy: 98.780488% Precision: 98.787789% Recall: 98.780488% F-Statistic: 98.782319% Confusion Matrix: [[162 2] [1 81]]	Accuracy: 98.373984% Precision: 98.373984% Recall: 98.373984% F-Statistic: 98.373984% Confusion Matrix: [[162 2] [2 80]]
Support Vector Machines (SVM)	Accuracy: 100.000000% Precision: 100.000000% Recall: 100.000000% F-Statistic: 100.000000%	Accuracy: 100.000000% Precision: 100.000000% Recall: 100.000000% F-Statistic: 100.000000% Confusion Matrix:	Accuracy: 99.593496% Precision: 99.595960% Recall: 99.593496% F-Statistic: 99.592867% Confusion Matrix:	Accuracy: 99.593496% Precision: 99.595960% Recall: 99.593496% F-Statistic: 99.592867% Confusion Matrix: [[164 0] [1 81]]

	Confusion Matrix: [[49 0] [0 25]]	[[52 0] [0 22]]	[[164 0] [1 81]]	
Logistic Regression	Accuracy: 98.648649% Precision: 98.676802% Recall: 98.648649% F-Statistic: 98.643012% Confusion Matrix: [[47 0] [1 26]]	Accuracy: 94.594595% Precision: 95.018548% Recall: 94.594595% F-Statistic: 94.488693% Confusion Matrix: [[47 0] [4 23]]	Accuracy: 99.593496% Precision: 99.595960% Recall: 99.593496% F-Statistic: 99.592867% Confusion Matrix: [[164 0] [1 81]]	Accuracy: 99.186992% Precision: 99.196787% Recall: 99.186992% F-Statistic: 99.184437% Confusion Matrix: [[164 0] [2 80]]

Table 4: Performance Metrics for Each Algorithm

Discussion

Implications

Despite the limited amount of data available, I thought the machine learning experiment went very well. All of the results achieved accuracies of at least 90 percent and many of the accuracies were near 100 percent. The lineup of the features by importance on Table 3 suggest that chronic conditions that a patient may have as well as basic info about the patient such as their birth date, gender or race, have meaning behind them that the computer was able to pick up. Towards the lower end of the rankings, are attributes that give monetary amounts, mostly information on claims payments and reimbursements. This indicates that they do not have as much of a say as the patient's conditions or characteristics. One thing that was surprising was the fact that the prescription drug information was in the middle of the rankings, indicating mediocre importance. Overall, running each feature selection algorithm gave very high importance rankings, indicating that most, if not all attributes have a considerable amount of importance and meaning.

All four machine learning algorithms proved to perform very well. However, the performances for each algorithm are different which can help distinguish the ranking of each algorithm. Support Vector Machines (SVM) performed the best with accuracies of 100 percent for both of its highest performing set of features and its most stable set of features for classic training and testing set. It is surprising that it reached 100 percent given the difficulty of the dataset. Having a smaller dataset may have served as an easier decision making platform for the computer to train and test on. The lowest performing algorithm was the Decision Tree and Logistic Regression with the greatest stability on the classic training and testing set, with an accuracy of 94 percent. Upon testing the optimal number of features for all four algorithms, I found that many of them performed the best with more features than less, with a few models using almost the entire feature set. This suggests that most if not all features play an important role in determining whether a patient is misusing prescription opioids.

As far as k-Fold Cross Validation goes, it performed very well and very consistently, maintaining its stability regardless of the number of selected features. The highest performing accuracies were given by the Decision Tree, Support Vector Machines (SVM), and Logistic Regression, all with accuracies of 99.5 percent. The lowest performing was still very impressive, given by Artificial Neural Networks (ANN) with an accuracy of 98.3 percent. Given the consistent exceptional performance of k-Fold Cross Validation, it suggests that it is less sensitive to noise in datasets and can still provide an accurate answer. On the other hand, the instability of the classic training and testing set suggests that it is more sensitive to noise and that having the optimal number of selected features does matter in order to achieve consistency. However, when optimized correctly, a higher performance can be achieved using a classic training and testing set.

After observing and analyzing my results, I have made a recommendation on what to include when creating the best model for predicting prescription opioid misuse. Since we are working with healthcare data on a federal level to be used to allocate federal spending and resources, it is important that we have a model that performs both exceptionally and consistently. Therefore, I have chosen Support Vector Machines (SVM) using k-Fold Cross Validation as the validation check. Although it's not perfect, it performs within one half of a percentage from 100 percent accuracy and it provides this level of accuracy almost every time. It was also able to achieve peak performance on almost the entire feature set, with the exception of the two lowest ranking features.

Limitations

There are several limitations that could potentially affect real world outcomes of this project. First, the full dataset included only inpatient and outpatient claims data over a three year span (2008 to 2010). It is possible for trends to change in later years due to changes and improvements in healthcare practices and technologies. Second, the final dataset was very small due to the small number of prescription opioid misuse cases. The total number of instances that indicated prescription opioid misuse was roughly 80. In order to provide a balance of data to make it easier for the computer to learn, I only included about twice as many negative instances, around 160. Therefore, the total number of instances was about 240. Although this did not appear to affect the results of this experiment, the results could differ using a larger dataset and could pose a need to make some adjustments to future models if a more robust set is used. In speaking of balancing data, the third limitation is that the balance of data was used for this experiment and does not provide an accurate portrayal of real-world data. The number of actual prescription misuse cases in the United States is very small and if my selected model was used in the real

world, the results could differ greatly. This is due to the learning process being much more difficult to achieve with a greatly imbalanced data set, especially having positive misuse cases in the single thousands versus having negative misuse cases in the millions. Lastly, my research focused on specifically looking at predicting cases of prescription opioid misuse on federal healthcare data provided by the Centers for Medicare and Medicaid Services (2013). Future researchers should be wary of using this information when applying it to other realms of healthcare subjects and to subjects outside of the healthcare industry.

Conclusion

This study has explored various machine learning algorithms in order to build the best model to be able to predict prescription opioids misuse in patients. This study also took this exploration further by experimenting with a dataset provided by the Centers for Medicare and Medicaid Services (2013). For the experiment portion, this study discovered the importance of each feature, tested four selected machine learning algorithms, and compared a classic training and testing set against k-Fold Cross Validation. From here, the study answered the questions of what is the best algorithm to use as well as the best features to use when creating the most ideal machine learning model. Upon analysis, it has been determined that Support Vector Machines using k-Fold Cross Validation would create the best model for predicting prescription opioid misuse.

Implementing a machine learning model derived from this study could have several possible outcomes. Brady et al. (2016) argued that many overdose deaths could be mitigated if healthcare and emergency personnel were equipped with an antidote such as Naloxone. This suggests that CMS could use these predictions and be able to allocate resources to different

communities where there is a high prediction of prescription opioid misuse. Other outcomes could include holding educational seminars to inform prescribed patients of the dangers of prescription opioids as well as allocating resources to affected communities to launch drug disposal programs (Brady et al. 2016). Cochran et al. (2017) also suggested that healthcare entities could hold interventions for patients who have been predicted to misuse prescription opioids in order to prevent further consequences from happening. After speaking with Kummet (2018), CMS would like to find communities and neighborhoods with high cases of potential misuse of prescription opioids to focus on providing resources such as antidotes, interventions, and educational programs as stated above.

With this in mind, I hope that this research project can make a contribution to the healthcare industry as well as communities affected by the prescription opioid crisis. Knowing who is misusing prescription opioids and even communities where many people are misusing them can help CMS and others know where to allocate resources in order to help alleviate the crisis. Having this technology at our fingertips will hopefully change the safety and quality of healthcare for the United States and the world for the better.

References

Brady, K. T., McCauley, J. L., Back, S. E. (2016). Prescription Opioid Misuse, Abuse, and Treatment in the United States: An Update. *Am J Psychiatry, 173*(1). pp. 18-26.

Centers for Medicare and Medicaid Services. (2013). Centers for Medicare and Medicaid Services (CMS) Linkable 2008 – 2010 Medicare Data Entrepreneurs' Synthetic Public Use File (DE-SynPUF) [Data file and user's manual]. Retrieved from https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF.html

Cochran, G., Gordon, A. J., Lo-Ciganic, W., Gellad, W. F., Frazier, W., Lobo, C., Chang, J., Zheng, P., Donohue, J. M. (2017). An Examination of Claims-Based Predictors of Overdose from a Large Medicaid Program. *Med Care, 55*(3). pp. 291-298.

Espino, J. U., Wagner, M., Szczepaniak, C., Tsui, F., Su, H., Olszewski, R., Liu, Z., Chapman, W., Zeng, X., Ma, L., Lu, Z., Dara, J. (2004). Removing a Barrier to Computer-Based Outbreak and Disease

Surveillance: The RODS Open Source Project. *Morbidity and Mortality Weekly Report, 53* (Supplement: Syndromic Surveillance, Reports from a National Conference, 2003). pp. 32-29.

Hosseinzaheh, A., Izadi, M., Verma, A., Precup, D., Buckeridge, D. (2013). *Assessing the Predictability of Hospital Readmission Using Machine Learning*. Paper presented at The Twenty-Fifth Innovative Applications of Artificial Intelligence Conference (pp. 1532-1538). Place of publication: Association for the Advancement of Artificial Intelligence.

Kaur, H., Wasan, S. K. (2006). Empirical Study on Applications of Data Mining Techniques in Healthcare. *Journal of Computer Science, 2*(2). pp. 194-200.

Koh, H. C., Tan, G. (2005). Data Mining Applications in Healthcare. *Journal of Healthcare Information Management, 19*(2). pp. 64-72.

Kummet, C. (2018, August 23). Personal interview.

Obenshain, M. K. (2004). Application of Data Mining Techniques to Healthcare Data. Infection Control and Hospital Epidemiology, 25(8). pp. 690-695.

Travis E. Oliphant. A guide to NumPy, USA: Trelgol Publishing, (2006).

Pedregosa, F., Veroquaux, A., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Duborg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). Sci-Kit Learn (0.11) [Computer Software] Retrieved from http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/index.htmls

Rose, S. (2018). Machine Learning for Prediction in Electronic Health Data. *JAMA Network Open, 1*(4). pp. 1-3.

Tomar, D., Agarwal, S. (2013). A Survey on Data Mining Approaches for Healthcare. *International Journal of Bio-Science and Bio-Technology, 5*(5). pp. 241-266.

Tzeng, H., Lin, Y., Hsieh, J. (2004). Forecasting Violent Behaviors for Schizophrenic Outpatients Using Their Disease Insights: Development of a Binary Logistic Regression Model and a Support Vector Model. *International Journal of Mental Health, 33*(2). pp. 17-31.

Vowles, K. E., McEntee, M. L., Julnes, P. S., Frohe, T., Ney, J. P., Van der Goes, D. N. (2015). Rates of Opioid Misuse, Abuse, and Addiction in Chronic Pain: A Systematic Review and Data Synthesis. *PAIN, 156*(4). pp. 569-576.

Wu, J., Roy, J., Stewart, W. F. (2010). Prediction Modeling Using EHR Data: Challenges, Strategies, and a Comparision of Machine Learning Approaches. *Medical Care, 48*(6). pp. S106-S113.